04 | 元数据中心的关键目标和技术实现方案

2020-04-10 郭忆

数据中台实战课 进入课程 >

讲述：郭忆
时长 20:02　大小 18.35M

你好，我是郭忆。

在上一节课程中，我从宏观的角度，系统性地带你了解了数据中台建设的方法论、支撑技术和组织架构，从这节课开始，我们正式进入实现篇，我会从微观的角度出发，带你具体分析数据中台的支撑技术，以电商场景为例，分别讲解元数据中心、指标管理、模型设计、数据质量等技术如何在企业落地。

这节课，咱们来聊聊元数据。

为什么要先讲元数据呢？我来举个例子。在原理篇中，我提到数据中台的构建，需要确保全局指标的业务口径一致，要把原先口径不一致的、重复的指标进行梳理，整合成一个统一的

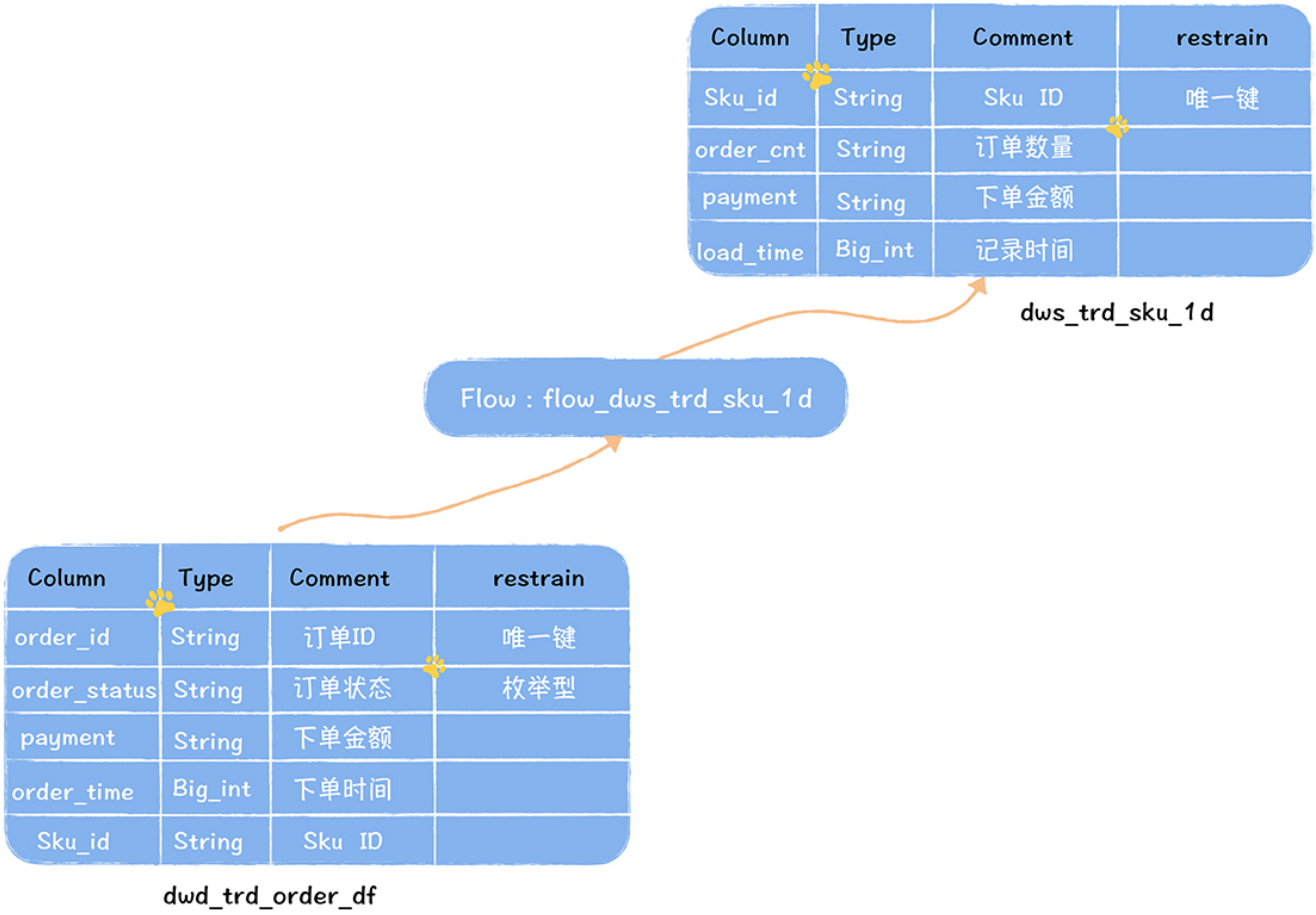
指标字典。而这项工作的前提，是要搞清楚这些指标的业务口径、数据来源和计算逻辑。而这些数据呢都是元数据。

你可以认为，如果没有这些元数据，就没法去梳理指标，更谈不上构建一个统一的指标体系。当你看到一个数 700W，如果你不知道这个数对应的指标是每日日活，就没办法理解这个数据的业务含义，也就无法去整合这些数据。**所以你必须要掌握元数据的管理，才能构建一个数据中台。**

那么问题来了：元数据中心应该包括哪些元数据呢？ 什么样的数据是元数据？

元数据包括哪些?

结合我的实践经验，我把元数据划为三类：数据字典、数据血缘和数据特征。我们还是通过一个例子来理解这三类元数据。



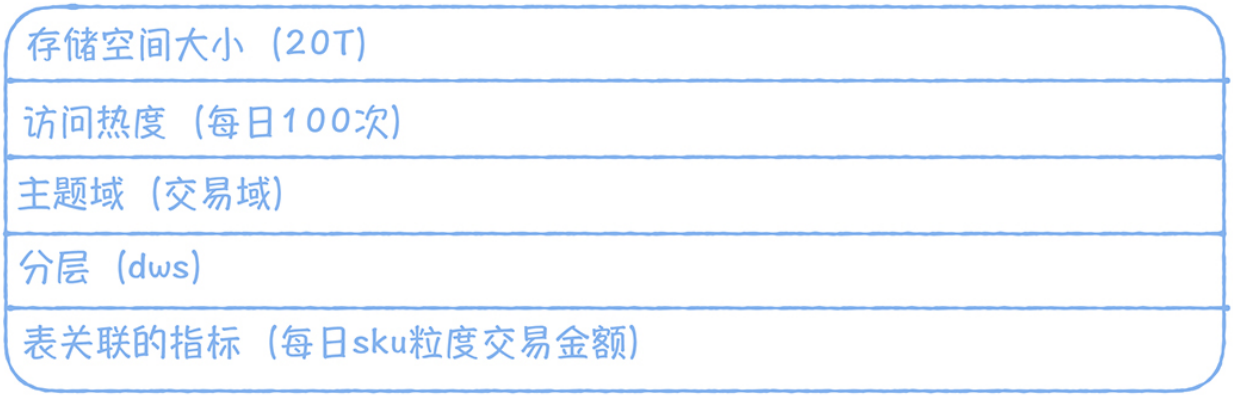
在这个图中，dwd_trd_order_df 是一张订单交易明细数据，任务 flow_dws_trd_sku_1d 读取这张表，按照 sku 粒度，计算每日 sku 的交易金额和订单数量，输出轻度汇总表 dws_trd_sku_1d。

数据字典描述的是数据的结构信息，我们以 dws_trd_sku_1d 为例，数据字典包括：

数据血缘是指一个表是直接通过哪些表加工而来，在上面的例子中，dws_trd_sku_1d 是通过 dwd_trd_order_df 的数据计算而来，所以，dwd_trd_order_df 是 dws_trd_sku_1d 的上游表。

数据血缘一般会帮我们做影响分析和故障溯源。比如说有一天，你的老板看到某个指标的数据违反常识，让你去排查这个指标计算是否正确，你首先需要找到这个指标所在的表，然后顺着这个表的上游表逐个去排查校验数据，才能找到异常数据的根源。

而数据特征主要是指数据的属性信息，我们以 dws_trd_sku_1d 为例：



通过这个例子，你了解了元数据了吗？ 不过元数据的种类非常多，为了管理这些元数据，你必须要构建一个元数据中心。那么接下来，我们就来看看如何搭建一个元数据中心，打通企业的元数据。

业界元数据中心产品

我做系统设计这么些年，一直有一个习惯，是先看看业界的产品都是怎么设计的，避免关门造车。业界的比较有影响力的产品：

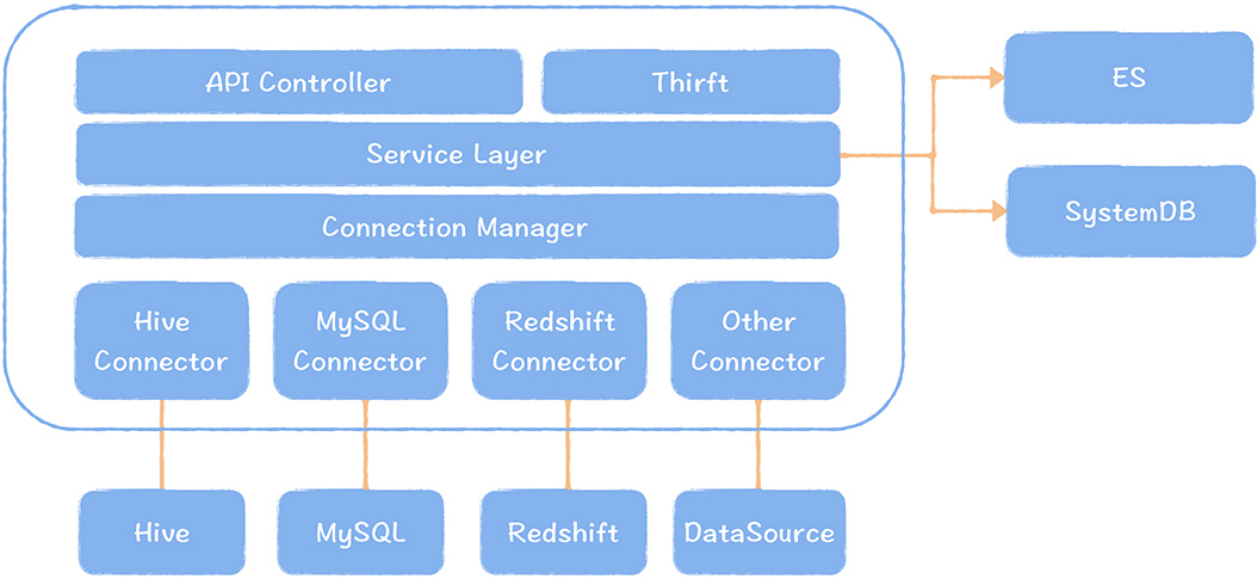
开源的有 Netflix 的 Metacat、Apache Atlas；

商业化的产品有 Cloudera Navigator。

我今天重点想带你了解 Metacat 和 Atlas 这两款产品，一个擅长于管理数据字典，一个擅长于管理数据血缘，通过了解这两款产品，你更能深入的理解元数据中心应该如何设计。

Metacat 多数据源集成型架构设计

关于Metacat，你可以在 GitHub 上找到相关介绍，所以关于这个项目的背景和功能特性，我就不再多讲，我只想强调一个点，就是它多数据源的可扩展架构设计，因为这个点对于数据字典的管理，真的太重要！



在一般的公司中，数据源类型非常多是很常见的现象，包括 Hive、MySQL、Oracle、Greenplum 等等。支持不同数据源，建立一个可扩展的、统一的元数据层是非常重要的，否则你的元数据是缺失的。

从上面 Metacat 的架构图中，你可以看到，Metacat 的设计非常巧妙，它并没有单独再保存一份元数据，而是采取直连数据源拉的方式，一方面它不存在保存两份元数据一致性的问题，另一方面，这种架构设计很轻量化，每个数据源只要实现一个连接实现类即可，扩展成

本很低，我把这种设计叫做集成型设计。我认为这种设计方式对于希望构建元数据中心的企业，是非常有借鉴意义的。

Apache Atlas 实时数据血缘采集

同样，关于Apache Atlas的背景和功能，我也不多说，只是想强调 Atlas 实时数据血缘采集的架构设计，因为它为解决血缘采集的准确性和时效性难题提供了很多的解决思路。

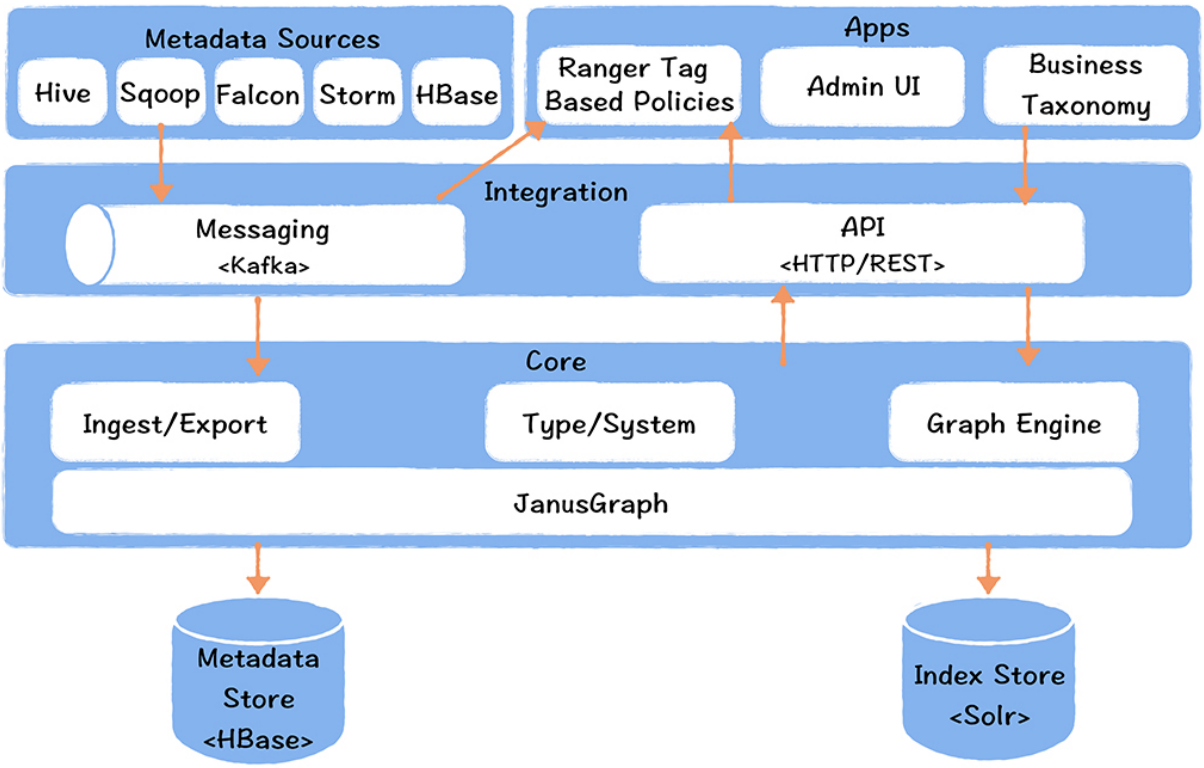
血缘采集，一般可以通过三种方式：

通过静态解析 SQL，获得输入表和输出表；

通过实时抓取正在执行的 SQL，解析执行计划，获取输入表和输出表；

通过任务日志解析的方式，获取执行后的 SQL 输入表和输出表。

第一种方式，面临准确性的问题，因为任务没有执行，这个 SQL 对不对都是一个问题。第三种方式，血缘虽然是执行后产生的，可以确保是准确的，但是时效性比较差，通常要分析大量的任务日志数据。所以第二种方式，我认为是比较理想的实现方式，而 Atlas 就是这种实现。



对于 Hive 计算引擎，Atlas 通过 Hook 方式，实时地捕捉任务执行计划，获取输入表和输出表，推送给 Kafka，由一个 Ingest 模块负责将血缘写入 JanusGraph 图数据库中。然后通过 API 的方式，基于图查询引擎，获取血缘关系。对于 Spark，Atlas 提供了 Listener 的实现方式，此外 Sqoop、Flink 也有对应的实现方式。

这两款产品在设计网易元数据中心时，给了很多灵感，下面我就带你了解一下网易元数据中心的设计，以便你掌握一个元数据中心在设计时应该考虑哪些点。

网易元数据中心设计

在设计网易元数据中心之初，我设定了元数据中心必须实现的 5 个关键目标：

其一，多业务线、多租户支持。

在网易，电商、音乐都是不同的业务线，同一个业务线内，也分为算法、数仓、风控等多个租户，所以元数据中心必须支持多业务线、多租户。

其二，多数据源的支持。

元数据中心必须要能够支持不同类型的数据源（比如 MySQL、Hive、Kudu 等），同时还要支持相同数据源的多个集群。为了规范化管理，还需要考虑将半结构化的 KV 也纳入元数据中心的管理（比如 Kafka、Redis、HBase 等）。这些系统本身并没有表结构元数据，所以需要能够在元数据中心里定义 Kafka 每个 Topic 的每条记录 JSON 中的格式，每个字段代表什么含义。

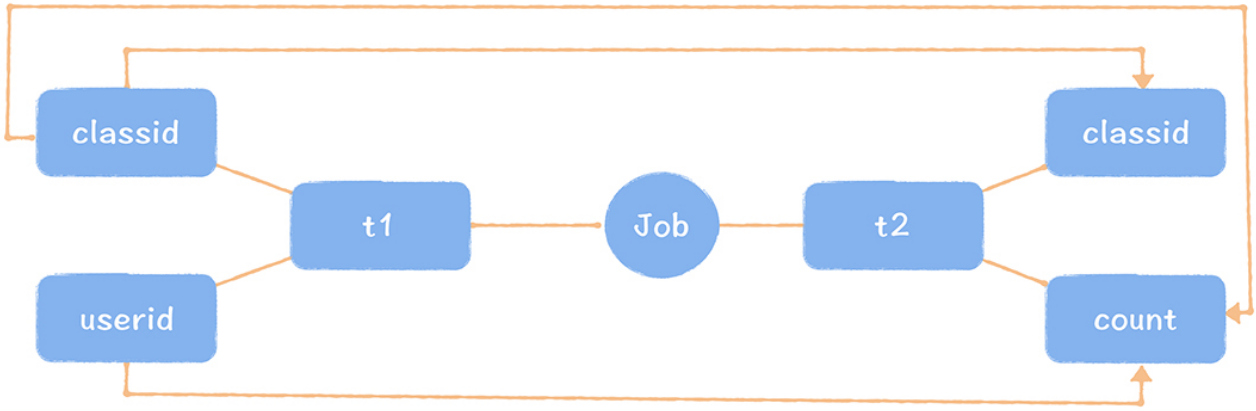
其三，数据血缘。

元数据中心需要支持数据血缘的实时采集和高性能的查询。同时，还必须支持字段级别的血缘。

什么是字段级别的血缘，我们来举个例子。

```
insert overwrite table t2 select classid, count(userid) from t1 group
by classid;
```

t2 表是由 t1 表的数据计算来的，所以 t2 和 t1 是表血缘上下游关系，t2 的 classid 字段是由 t1 的 classid 字段产生的，count 字段是由 userid 经过按照 classid 字段聚合计算得到的，所以 t2 表的 classid 与 t1 的 classid 存在字段血缘，t2 表的 count 分别与 t1 表的 classid 和 userid 存在血缘关系。



字段血缘在做溯源的时候非常有用，因为大数据加工链路的下游是集市层，为了方便使用者使用，一般都是一些很宽的表（列很多的表，避免 Join 带来的性能损耗），这个表的上游可能是有几十个表产生的，如果不通过字段血缘限定溯源范围，就会导致搜索范围变得很大，无法快速地精准定位到有问题的表。

另外，数据血缘还必须要支持生命周期管理，已经下线的任务应该立即清理血缘，血缘要保留一段时间，如果没有继续被调度，过期的血缘关系应该予以清理。

其四，与大数据平台集成。

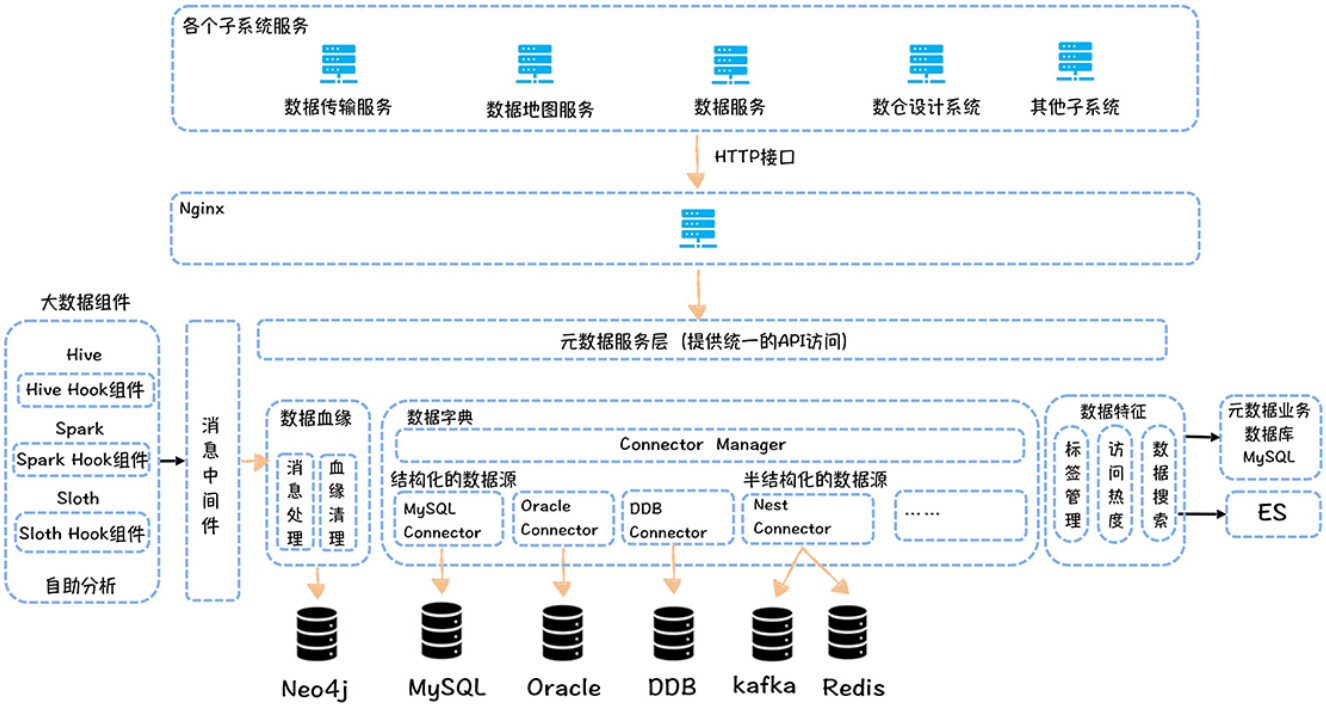
元数据中心需要与 Ranger 集成，实现基于 tag 的权限管理方式。在元数据中心中可以为表定义一组标签，Ranger 可以基于这个标签，对拥有某一个标签的一组表按照相同的权限授权。这种方式大幅提高了权限管理的效率。比如，对于会员、交易、毛利、成本，可以设定表的敏感等级，然后根据敏感等级，设定不同的人有权限查看。

另外，元数据中心作为基础元数据服务，包括自助取数分析系统，数据传输系统，数据服务，都应该基于元数据中心提供的统一接口获取元数据。

其五，数据标签。

元数据中心必须要支持对表和表中的字段打标签，通过丰富的不同类型的标签，可以完善数据中台数据的特征，比如指标可以作为一种类型的标签打在表上，主题域、分层信息都可以作为不同类型的标签关联到表。

基于这 5 个因素的考虑，我们设计了网易元数据中心。



网易元数据中心系统架构设计图

这个图按照功能模块分为数据血缘、数据字典和数据特征。

数据血缘由采集端、消息中间件、消费端以及血缘清理模块组成，基于 Hive Hook，Spark Listener，Flink Hook ，可以获取任务执行时输入表和输出表，推送给统一的消息中间件（Kafka），然后消费端负责将血缘关系沉淀到图数据库中。

图数据库选择 Neo4j，主要考虑是性能快、部署轻量化、依赖模块少，当然，开源的 Neo4j 没有高可用方案，并且不支持水平扩展，但是因为单个业务活跃的表规模基本也就在几万的规模，所以单机也够用，高可用可以通过双写的方式实现。

血缘还有一个清理的模块，主要负责定时清理过期的血缘，一般我们把血缘的生命周期设置为 7 天。

数据字典部分，我们参考了 Metacat 实现，我们由一个统一的 Connector Mananger 负责管理到各个数据源的连接。对于 Hive、MySQL，元数据中心并不会保存系统元数据，而是直接连数据源实时获取。对于 Kafka、HBase、Redis 等 KV，我们在元数据中心里内置了一个元数据管理模块，可以在这个模块中定义 Value 的 schema 信息。

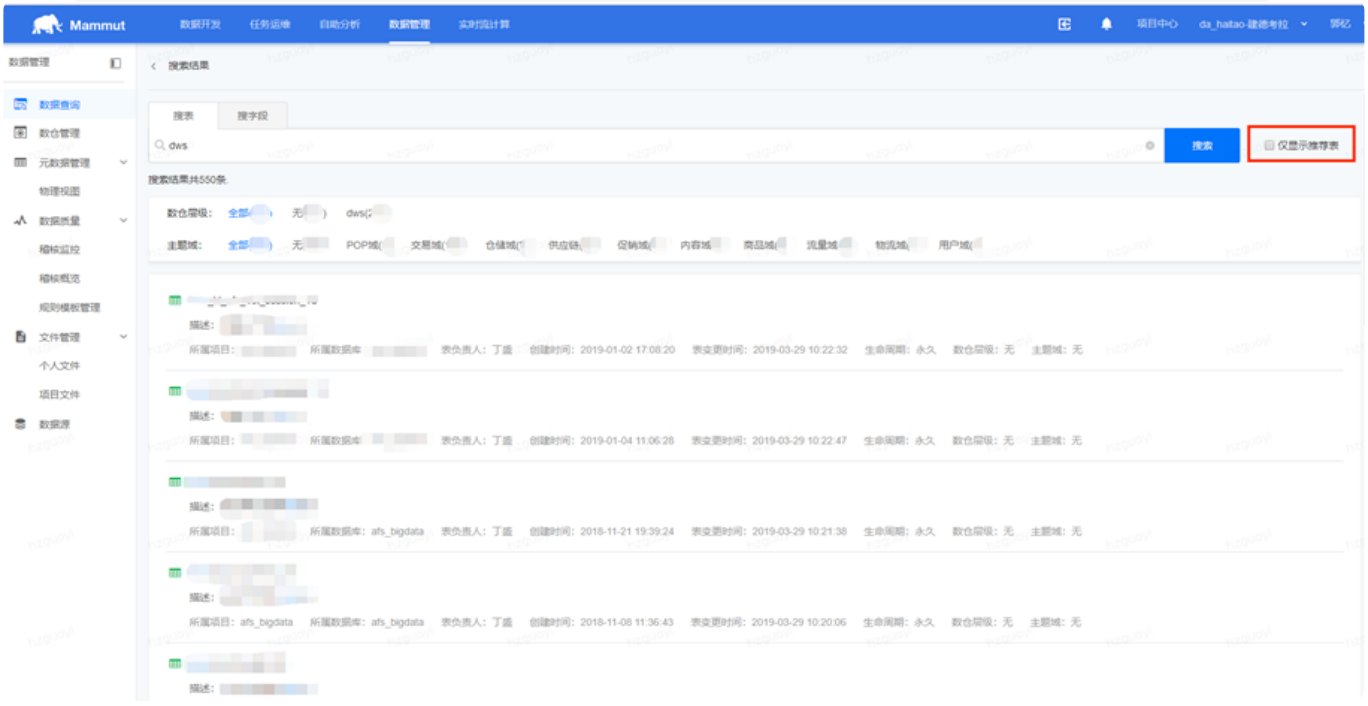
数据特征主要是标签的管理以及数据的访问热度信息。元数据中心内置了不同类型的标签，同时允许用户自定义扩展标签类型。指标、分层信息、主题域信息都是以标签的形式存储在元数据中心的系统库里，同时元数据中心允许用户基于标签类型和标签搜索表和字段。

元数据中心统一对外提供了 API 访问接口，数据传输、数据地图、数据服务等其他的子系统都可以通过 API 接口获取元数据。另外 Ranger 可以基于元数据中心提供的 API 接口，获取标签对应的表，然后根据标签更新表对应的权限，实现基于标签的权限控制。

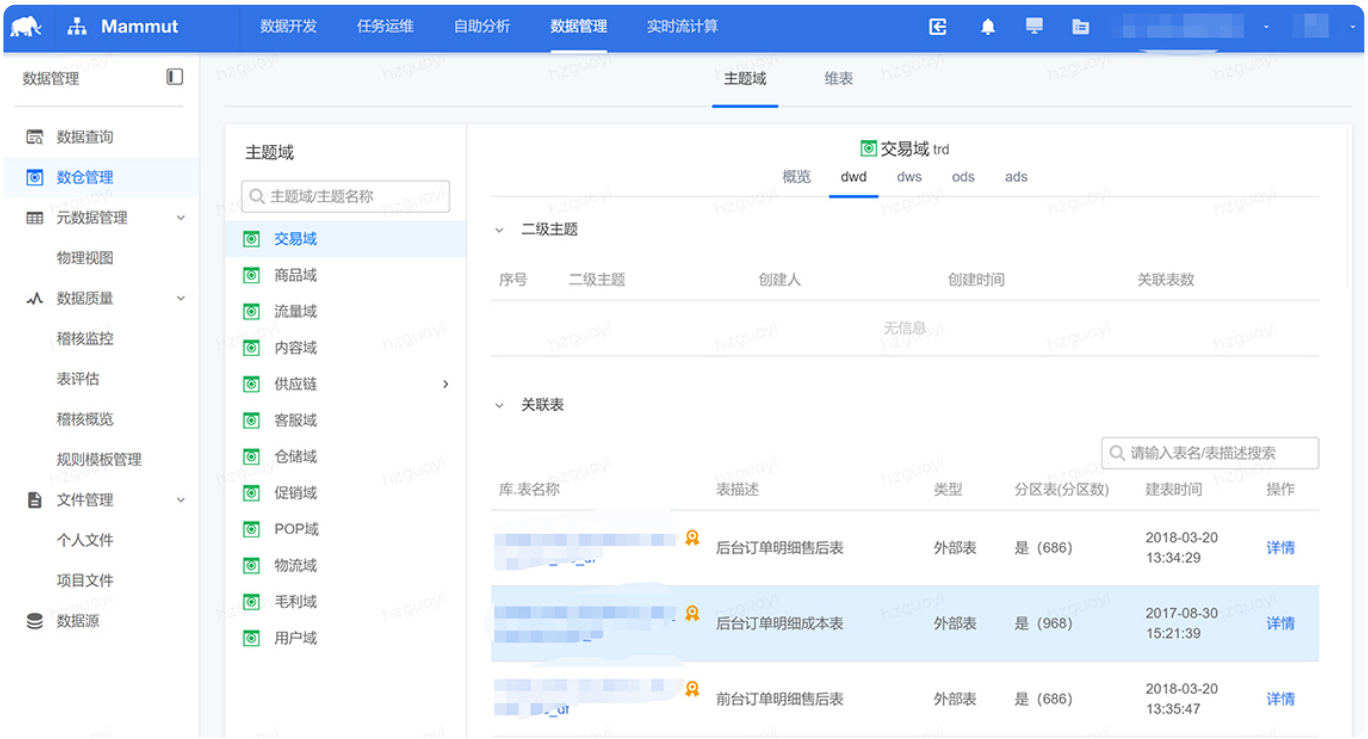
元数据中心构建好以后，你肯定会问，这个元数据中心没有界面吗？它长什么样子？用户咋使用这个元数据中心？ 别急，我们接着往下看。

数据地图：元数据中心的界面

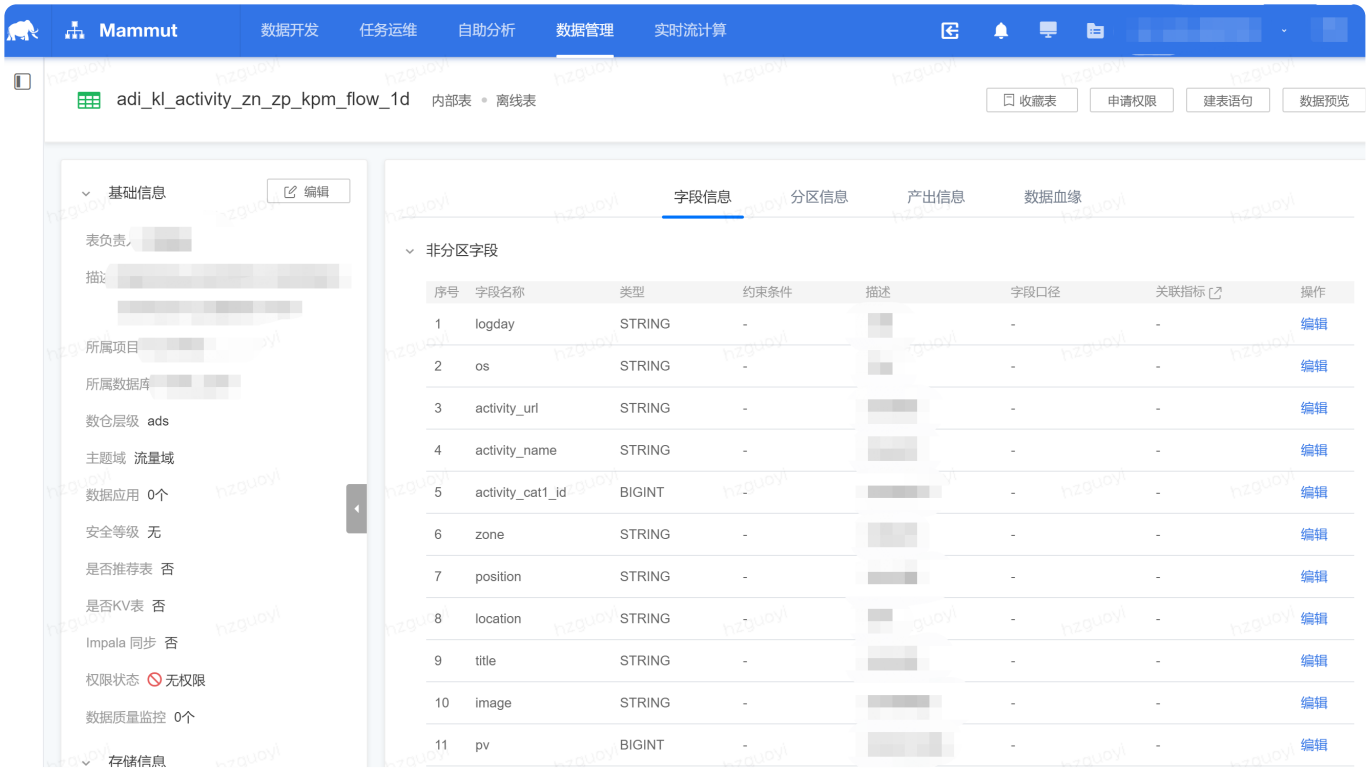
数据地图是基于元数据中心构建的一站式企业数据资产目录，可以看作是元数据中心的界面。数据开发、分析师、数据运营、算法工程师可以在数据地图上完成数据的检索，解决了“不知道有哪些数据？” “到哪里找数据？” “如何准确的理解数据”的难题。



数据地图提供了多维度的检索功能，使用者可以按照表名、列名、注释、主题域、分层、指标进行检索，结果按照匹配相关度进行排序。考虑到数据中台中有一些表是数仓维护的表，有一些表数仓已经不再维护，在结果排序的时候，增加了数仓维护的表优先展示的规则。同时数据地图还提供了按照主题域、业务过程导览，可以帮助使用者快速了解当前有哪些表可以使用。



当使用者定位到某一个表打开时，会进入详情页，详情页中会展示表的基础信息，字段信息、分区信息、产出信息以及数据血缘。数据血缘可以帮助使用者了解这个表的来源和去向，这个表可能影响的下游应用和报表，这个表的数据来源。



数据地图同时还提供了数据预览的功能，考虑到安全性因素，只允许预览 10 条数据，用于判断数据是否符合使用者的预期。数据地图提供的收藏功能， 方便使用者快速找到自己经常使用的表。当数据开发、分析师、数据运营找到自己需要的表时，在数据地图上可以直接发起申请对该表的权限申请。

数据地图对于提高数据发现的效率，实现非技术人员自助取数有重要作用。经过我的实践，数据地图是数据中台中使用频率最高的一个工具产品，在网易，每天都有 500 以上人在使用数据地图查找数据。

课程总结

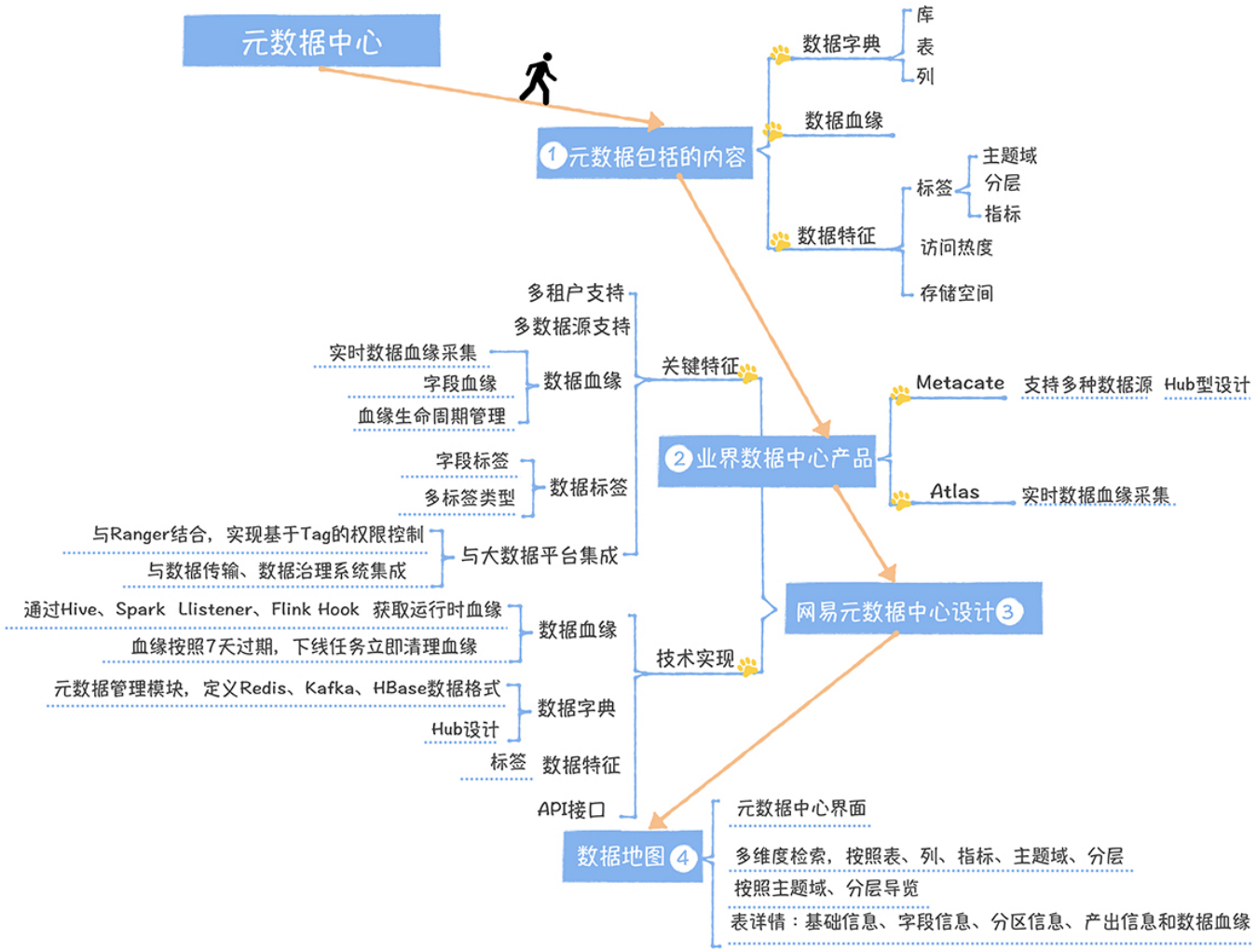
本节课，我以元数据作为起点，带你了解了元数据应该包括数据字典、数据血缘和数据特征，然后通过分析两个业界比较有影响力的元数据中心产品，结合我在网易数据中台实践，给出了元数据中心设计的 5 个关键特性和技术实现架构，最后介绍了基于元数据中心之上的数据地图产品。我想在最后强调几个关键点：

元数据中心设计上必须注意扩展性，能够支持多个数据源，所以宜采用集成型的设计方式。

数据血缘需要支持字段级别的血缘，否则会影响溯源的范围和准确性。

数据地图提供了一站式的数据发现服务，解决了检索数据，理解数据的“找数据的需求”。

最后，你要知道，元数据中心是数据中台的基石，它提供了我们做数据治理的必须的数据支撑，在后续的章节中，我们将逐一介绍指标、模型、质量、成本、安全等的治理，这些都离不开元数据中心的支撑。



课程思考

在课程中，我介绍了血缘采集的三种方式，并且推荐了通过实时采集的方式，但是其实静态解析血缘也有它的优势应用场景，你能想到有哪些么？欢迎在留言区与我讨论。

最后，感谢你的阅读，如果这篇文章让你有所收获，也欢迎你将它分享给更多的朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　特别放送 | 史凯：建设数据中台到底有什么用?

下一篇　05 | 如何统一管理纷繁杂乱的数据指标?

精选留言 (17)

写留言

惜心（伟祺）
2020-04-10

业务指标
数据来源
加工sql
把数据生命周期当作产品服务，提供给公司人员使用
和公司把具体产品提供给外面实验是一个思路...
展开

作者回复: 对的，其实思路是一致的，数据产品可以看成是一个C端产品，它的客户不是开发，而是运营，所以在产品设计上，要尽可能的降低门槛，注重引导。

1　3

麻婆豆腐
2020-04-13

郭老师好，听了您的课，感觉个个都直击痛点啊！本节课里数据字典和数据血缘感觉都有开源的可以参考或者直接使用，那么数据特征的管理是怎么处理的呢？手动维护吗？比如标签、关联指标之类描述性的。

1

张战文
2020-04-11

郭老师我听了您数据中台实战课，我们单位有需求且量比较大，能军民融合一下么？特别需要您的指点

作者回复: 好啊，数据中台与业务结合更紧密，具备行业属性，因为网易业务比较多元化，所以我们做过电商、在线音乐、新闻、在线教育等行业，对外也做过零售、物流、农业、制造业等行业，希望能够有更多的行业实践~

1 1

Galen
2020-04-11

感觉一般的小团队，搞不定啊

展开

作者回复: 你好，其实不然， 元数据中心业界有开源的产品，其实最差也可以用开源的来搭一套，只是没有那么易用罢了。元数据中心本身还是一个偏实现层的产品，基于元数据中心之上，我会为你介绍五个元数据的应用场景，这部分开源的产品会比较少涉及，但是如果你能深入理解这些产品背后设计思想，应用场景，解决的问题，即使你要选取外面的商业化产品，你也可以有自己的一个判断。

感谢你的阅读，期待与你在留言区再次相遇~

1

Marco
2020-04-10

老师，如果表数据是通过java 程序的etl，又如何解析血缘关系？

作者回复: 目前，我们数据中台中所有的数据都是以表的形式存在的，血缘都是以表的血缘。并没有做文件、数据集的血缘。

感谢你的阅读，期待与你在留言区再次相遇。

1

张振华
2020-04-10

表字段信息是实时采集的，像表负责人这些信息怎么关联上的?

作者回复: metastore是有owner属性的，它可以作为表的负责人角色。

另外，对于非hive表，负责人可以作为一种类型的标签，和表建立关联。

期待与你下一次在留言区互动~

1 1

Samuel
2020-04-10

郭老师,您好,元数据中心建设,是否可以理解主要以元数据管理工具进行落地,只是需要配置,就可以实现呢?还是需要有相关的代码开发的工作,才能落地元数据中心的建设?

作者回复: 元数据中心的建设，对于数据字典中直连数据源获取元数据的数据源，以及数据血缘部分，工具落地就可以统一收集到元数据。

但是对于数据特征，尤其是指标、维度标签，这部分是需要数据开发实施介入的，需要进行规范化梳理，一个表，哪些字段是指标，哪些字段是维度，这些不是工具落地就可以自动获取的。

感谢你的阅读，期待与你在留言区再次互动~

1

Silence L
2020-04-13

郭老师好，看了元数据我又两个问题请教一下：
1.文中元数据中心依赖了Atlas，ranger，neo4j，es，kafka等，是否依赖的太多，太重
2.ranger通过tag实现权限管理，是否数据权限管理都使用ranger，不会另外单独一个数据权限模块么?

...

展开

杨柳

2020-04-13

网易的元数据中心是完全自研的吗？中小型企业自建元数据中心是推荐在MetaCat或者Atlas上改还是自研呢？

lansane

2020-04-12

郭老师，你好，就元数据的存储模型，是否能分享下经验？之前看了普元专门做元数据管理的公司，说底层都是兼容cwm模型

展开

Robbin

2020-04-11

在传统企业里，高层领导都是业务出身，而像元数据中心这种产品，如何能说服业务领导同意建设，同时数据地图在设计时如何能让纯业务人员感受到其价值？

展开

作者回复: 你好，你说的很对，元数据中心本身是一个偏实现层的产品，领导其实根本就不关心是否存在这样的一个数据中台的底层。

但是数据地图，是元数据中心的界面，通过数据地图，领导可以看到数据中台的统一元数据视图，另外，结合数据地图的使用频率、使用范围，可以凸显数据地图的价值。

数据地图在设计时，一方面他的使用对象是数据开发，另外一方面，他的使用对象又是业务人员。让业务人员感受到数据地图的价值，主要是能够让业务人员搜索指标、数据报表，帮助他们快速找到自己想要的数据。无论是数据表，还是数据报表，还是指标，都能够通过数据地图进行搜索和导览。

Robbin

2020-04-11

对于静态数据结构、动态血缘分析这种可以通过工具采集，但是数据库中没有中文，是否还是需要人工梳理录入登记？这个工作量也不小了。

展开

作者回复: 你好，你是指的数据字典中字段级别的业务元数据信息？ 有的数据库中，并没有相关的commet描述，或者comment 不适合查看，此时可以通过标签的形式，作为一种特定类型的标签，关联到表的字段中。

本身这个梳理的工作是跑不掉的，但是也可以采取用到的时候再补充，数据源是数据集成阶段登记到数据中台的元数据中心中，此时再梳理补充，并不需要一口气全部补充完整。

aof
2020-04-11

郭老师，您觉得不同规模的企业，构建一个数据中台大概需要多少人手和多久时间？

作者回复: 你好，其实这个问题，没有办法给出一个统一的标准答案，因为不同企业的业务复杂度，数据应用的水平、建设水平差异都很大。

我会在专栏第13节介绍网易电商数据中台项目从立项到实施的完成过程，可以给你一定的参考。

但是我要多说一句，既然问到这个问题，很可能就是你在担心数据中台建设，前期需要投入很多的资源。而我的建议是，数据中台的建设可以采取滚雪球的方式，逐步以场景化的方式落地。这样既可以控制前期的资源投入风险，又可以保证数据中台有一些阶段性成果的输出。

感谢你的阅读，期待你在留言区再次与我互动~

summer
2020-04-10

静态解析可以在任务开始之前提供给SQL开发者一些信息，如：上游表有问题、当前资源不足、已存在类似任务等信息，避免错误和资源浪费。
另外，提前统计也可以让数据中台开发者对任务量级有个预估

展开

作者回复: 其实，主要是解决SQL还没有执行过，没有血缘产生，而此时又需要用到血缘的场景~

感谢你的阅读，期待与你在留言区再次相遇~

咸蛋猫熊
2020-04-10

请问下，元数据中心在项目运维中是怎么和调度系统结合的？ 比如表的使用热度等信息是基于什么指标进行判断的？

展开

作者回复: 你好，表的使用热度，是根据平台上调度运行的job和adhoc执行的query计算来的。

通过数据血缘，我们可以获取到表和任务、query的关联关系，然后可以计算这部分的引用热度。数据血缘是通过hive/spark插件的方式获取的。

感谢你的交流，期待与你在留言区再次相遇~

Terry郑
2020-04-10

这一期录音频真是辛苦老师了，哈哈。
老师有一个问题想请教下，元数据的初始真实性您在实战中是如何解决的呢。

展开

作者回复: 你好，你所指的真实性是指什么？

能有一个具体的例子么？

因为在文章中，我也提到元数据中心，管理了所有数据中台的元数据，所有系统都与元数据中心打通，把元数据的管理入口都收归到元数据中心，可以确保元数据的一致。

期待与你再次在留言区相遇~

JohnT3e
2020-04-10

我能想到的一个场景是：静态血缘解析可以对一个正在开发的SQL提供参考信息，看系统中表有哪些SQL处理，避免SQL冗余和冲突。

展开

作者回复: 我来举个场景，你来看看。

当我们要提交任务上线，建立任务依赖时，如果我们依赖的表，还没有被调度产生数据，此时就会导致我们根据这张表找不到表的产出任务，系统就无法自动推荐依赖任务。

所以此时就需要静态血缘的介入啦。 对于还未执行，但是保存，SQL语法检查通过的任务，我们可以通过解析SQL获取静态血缘，然后当其他任务读取这张表，要建立到这张表产出任务的依赖时，我们可以根据静态血缘，找到这张表的产出任务。

欢迎你继续在留言区与我互动~